Method for Supporting User Level Online Diagnostics on Linux

BACKGROUND OF THE INVENTION

Technical Field

This invention relates to a module for supporting online diagnostics in a Linux operating system. More specifically, the module enables mapping of contiguous physical memory from kernel address space to a user level address space.

Description Of The Prior Art

Conventional memory allocators return memory which is contiguous in the virtual address space but is not guaranteed to be contiguous in the physical address space. For example, the Linux operating system's user level kernel memory allocator (malloc) does not allocate memory blocks which are guaranteed to be contiguous in the physical address space. Accordingly, memory allocated using conventional mechanisms are not guaranteed to be usable for online cache testing in the Linux operating system environment.

In prior art computer systems utilizing the Unix operating system, a special memory buffer is allocated during kernel initialization to allow online processor cache testing . The memory buffer is mapped from the kernel level to the user level program. The buffer accounts for the requirement of allocating memory returns that are contiguous in both the virtual address space and the physical address space. However, the prior art memory buffer requires modification of the kernel by allocating the memory buffer during kernel initialization. In addition, the prior art memory buffer allocated at initialization remains in the system even when it is not required. Accordingly, the prior art memory buffer for the Unix operating system is static and does not allow active allocation of memory on an as needed basis.

There is therefore a need for a module that will allow user level processes to generate memory access which can be used to test the hardware caches in a Linux operating system on an

as needed basis. The module must return memory to the user level which is contiguous in both the virtual and physical address spaces. Accordingly, it is desirable to provide a module that allows online testing of hardware cache without changing the kernel of the operating system.

SUMMARY OF THE INVENTION

5   This invention comprises a module for supporting online cache diagnostics in a Linux operating system. The module is dynamically loadable from the user space.

A first aspect of the invention is a method for supporting memory diagnostics. A module is loaded to a kernel of the operating system. A block of contiguous memory is allocated. Contiguous memory is mapped from the kernel address space to the user process address space.

10   Control of the physical memory displacement between accesses is controlled from the user level. The memory is contiguous in both physical and virtual address spaces.

A second aspect of the invention is a computer system which includes a module adapted to be dynamically loaded to a kernel of an operating system. The module includes a manager to allocate contiguous memory. The manager maps contiguous physical memory from a kernel

15   address space to a user process address space. The memory is contiguous in both the physical and virtual address space. A user level program controls physical memory displacement between accesses.

A third aspect of the invention is an article comprising a computer-readable signal bearing medium. In the medium, the article includes a module adapted to be dynamically loaded

20   to a kernel of an operating system for allocating a block of contiguous memory. The module includes a manager for mapping contiguous memory from a kernel address space to a user process address space. A user level program controls physical memory displacements between accesses.

Other features and advantages of this invention will become apparent from the following detailed description of the presently preferred embodiment of the invention, taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

5   FIG. 1 is a block diagram illustrating placement of the module within the Linux kernel.

FIG. 2 is a block diagram illustrating mapping from the kernel to a user process space according to the preferred embodiment of this invention, and is suggested for printing on the first page of the issued patent.

FIG. 3 is a block diagram illustrating mapping from the kernel to multiple user processes.

10   ## DESCRIPTION OF THE PREFERRED EMBODIMENT

Overview

Most known schemes for mapping hardware cache lines include using part of the physical address space and require the memory used in that physical address space to be physically contiguous. In addition, in order to preserve relative displacements between memory locations,

15   the memory used should also be contiguous in the virtual address space. Memory which is contiguous in the virtual address space may not be contiguous in the physical address space. The method and system of the preferred embodiment employ a module adapted to be loaded to the kernel to allow user level programs to conduct online diagnostics of cache hardware using physically and virtually contiguous memory while the system is operating under a normal

20   workload.

Technical Background

The Linux kernel has a Kernel Memory Allocator which always returns memory which is contiguous in both the virtual and physical address spaces. However, user level programs do not have access to the Kernel Memory Allocator. By designing a diagnostic tool that allows the user

level the ability to access the Kernel Memory Allocator, the requirement that contiguous lines of memory in both the virtual and physical address spaces be provided for diagnostic tools requiring such contiguous virtual and physical memory can be met through accesses by user processes.

Linux operating systems are designed to accommodate modules. A module functions as
5 an extension to the kernel and mirrors a device driver without accessing hardware, *i.e.* a pseudo device driver. During use, the module is accessed using device characteristics.

The module 20 of the preferred embodiment of the invention is loaded to the kernel level of the operating system and enables user level programs to have access to the Kernel Memory Allocator. The module does not modify the kernel. Rather, the module is available for activation
10 from the user space.

The module is specifically designed to allow online cache diagnostic testing. When the module is not activated it remains in the kernel without performing any mapping functions. Fig. 1 is an illustration of the module in an inactive state. The module 20 is loaded into the kernel level 40 of the operating system. A kernel memory buffer 30 is allocated when the module 20 is
15 loaded. The buffer is allocated even when diagnostics are not being conducted. The user level 60 has an area 80 designated for conducting online cache diagnostics. However, in this illustration there are no diagnostics being conducted, and as such the kernel memory buffer is not mapped to the diagnostic process's user space.

The module 20 is designed to be accessed from the user level 60. Fig. 2 is an illustration
20 of the module in an active state. The module 20 is accessed by a computer operator in the user level 60. A user process initiating an online cache diagnostic executes an open operation (open) followed by a memory map operation (mmap) from the user level. This causes the module to map a contiguous kernel memory buffer 30 from the kernel level 40 to the user level memory buffer 70. The online cache diagnostic uses the mapping to generate accesses to contiguous
25 lines of virtual and physical memory. Since the buffer is contiguous in both the physical and virtual address spaces, the online diagnostic can control the physical memory displacements
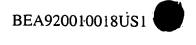
between accesses. In addition, the control of the physical memory displacements enables oversight and enforcement of specific cache operations. The memory buffer 30,70 is only mapped when accessed from the online diagnostic 80 in the user level 60. Once the online diagnostic is completed, the mapping is discontinued. However, the module 20 does remain in

5   the kernel. Accordingly, the module enables mapping of a memory buffer from the kernel to the user level on an as needed basis.

The module 20 shown in Fig. 2 is an illustration of a single user process accessing the contiguous kernel memory buffer. However, the same module 20 is adapted to map to multiple user processes concurrently. Fig. 3 is an illustration of two user processes utilizing the online

10   cache test support module. Similar to a single user process, a second mapping 70a of contiguous kernel memory buffer 30 is allocated from the kernel 40 to the user space 60a. This same kernel memory buffer 30 is mapped to the user process of all the user level programs which have access to the module 20, thereby enabling testing of inter-processor cache coherency features. Accordingly, processes running on different processors may access the same physical memory.

15   In a further embodiment, this module may be adapted to processors and systems which have different caching structures. The module 20 provides a generic mechanism for mapping contiguous memory from the kernel level 40 to the user level 60. The module 20 provides support for testing all standard memory caching hardware. The memory buffer is created when the module is accessed from the user space. The detailed knowledge required to test the cache is

20   contained in the user level test program and not in the module 20. Accordingly, the module 20 is merely a tool for enabling cache testing on any hardware platform.

Advantages Over The Prior Art

The module of the preferred embodiment enables user level processes to generate memory accesses which can be used to test the hardware caches. The buffer created by the

25   module is contiguous in both the virtual and physical memory address space. Since the memory being accessed is contiguous, cache replacement line algorithms can be tested at the same time.

In addition, the module does not require modification to the kernel. Similar to a driver, the module is dynamic and only functions when it is accessed from the user level. Accordingly, the module functions as a pseudo driver that is loadable to the kernel subsequent to initialization, and it enables diagnostic testing of cache hardware from the user level in a dynamic mode.

5

Alternative Embodiments

It will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. In particular, the module may be designed to allocate memory buffers upon request. The memory buffer would only be present upon 10 activation from the module, being deallocated upon deactivation of the module. Therefore system resources would be minimized when they are not active. Accordingly, the scope of protection of this invention is limited only by the following claims and their equivalents.